HATE SPEECH DETECTION USING DEEP LEARNING

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

B.Tech.

by

Anurag Srivastava (2018IMT-019)

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 010

2021

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **Hate Speech Detection using Deep Learning**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *June 2021* to *october 2021* under the supervision of **Dr. Vinal Patel** and **Dr. Santosh Singh Rathore**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Anurag
11/08/21

Date: Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisors

ABSTRACT

Via online media, disdain discourse as bigoted and misogynist articulations is all around inescapable. As a result, several social media platforms address the issue of detecting hate speech. Hate speech identification on text is essential for applications such as extracting controversial events, creating AI chatterbots, content recommendation, and sentiment analysis. This task is defined as the ability to classify a text as hate speech or not. This endeavour is difficult due of the intricacy of natural language constructs. In this work, dataset used is tweets using twitter API, different models were analyzed but CNN seems to be most promising one because it has not do deal specifically with text as language but as structure to determine Hate speech or not.

Keywords: Sliding Window · Hate Speech · Bag-Of-Words · Convolutional NN · Deep Learning · NLP · Word Embedding · Offensive Content · Hindi

ACKNOWLEDGEMENTS

I am highly indebted to **Dr. Vinal Patel** and **Dr. Santosh Singh Rathore**, and obliged for giving me the autonomy of functioning and experimenting with ideas. I would like to take this opportunity to express my profound gratitude to them not only for their academic guidance but also for their personal interest in my project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me. The nurturing and blossoming of the present work is mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. My mentor always answered myriad of my doubts with smiling graciousness and prodigious patience, never letting me feel that I am novices by always lending an ear to my views, appreciating and improving them and by giving me a free hand in our project. It's only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, I am grateful to our Institution and colleagues whose constant encouragement served to renew my spirit, refocus our attention and energy and helped my in carrying out this work.

Anurag

(Anurag Srivastava)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND RELATED WORK

This chapter gives brief introduction of the project as well as literature survey and related works done over the task.

1.1 INTRODUCTION

1.1.1 Hate Speech Detection

Hate Speech is an immediate or roundabout assertion focused on an individual or gathering of individuals proposed to belittle and mistreat another or utilize deprecatory language dependent on nationality, religion, incapacity, sex, or sexual direction. Hate speech has continuously increased as a result of the tremendous increase in user-developed content on social media. Hate speech directed towards a specific person or group of individuals can result in personal anguish, cyberbullying, societal panic, and discrimination. In reaction to the rise in hate speech on social media, a few of studies on automatic hate speech detection have been published in an attempt to reduce online harassment. While some relevant and independent work on code-mixed social media content exists, few attempts to detect hate speech in Hinglish data have been performed. We have unlabelled tweets and the method here depends on pre-training word vectors using a compatible collection of the same tweets dataset and training CNN (convolutional Neural Networds) model in conjunction with the labelled training set.

1.2 MOTIVATION

In a study conducted in USA [link] , while utilising social media, 12% of students said they frequently saw racist hate speech. Overall, 52% of respondents said they saw racist hate speech on social media on a regular or irregular basis. With the increase in data all over, most of which remains unchecked and unregulated, hate speech is quite prevalent. The available mechanisms to monitor these things are mostly modeled for the English language. For a country like India, where people are multilingual, hate speech detection becomes even challenging. With this motivation, we here try to extend the reach of hate speech detection to Hindi and Hindi-English mixed text.

1.3 LITERATURE REVIEW

Significant early work on hate-speech recognition was done by Spertus (1997), who constructed a model framework Smokey utilizing a C4.5 choice tree generator to decide highlight-based guidelines that could order harmful messages. From that point forward, disdain discourse identification has accomplished achievements, and a few models have been prepared to recognize disdain discourse. Yin et al. (2009) were quick to utilize a regulated learning way to deal with distinguish provocation on web 2.0. They grouped online media posts utilizing a support-vector machine (SVM) in light of nearby relevant and assessment features. Zampieri and Malmasi (2017) [1] inspected character n-grams, word n-grams and skip-grams to recognize disdain discourse in web-based media. They prepared their classifier on an English informational index with three marks and accomplished an exactness of 78%.

Hate speech detection on web platforms has been made possible because to advances in ML and NLP. ML and Deep Learning approaches for automated-hate-speech and offensive material detection have been the subject of numerous scientific studies. Character level and word level n-grams are the commonly used features in ML based approaches, and so on Albeit directed ML based methodologies have utilized diverse content mining-based components like surface provisions, assumption examination, lexical resources,linguistic highlights, information based elements, or client based and stage based metadata, they require a distinct element extraction approach. These days, the neural-network models apply text representation and deep learning approaches such as CNN [2], Bi-directional Long-Short-Term Memory Networks (LSTMs) [3], and Bidirectional Encoder Representations from Transformers [4] to improve the performance of hate speech detection models.

1.4 OBJECTIVE

The main objective is to design a deep learning model that is capable of classifying given text as hate speech or not. This is essentially a binary classification problem. We concentrate on Twitter, which is the most extensively utilised data source in the study of abusive language. We use all publicly available datasets with tweets categorised as various sorts of abuse and written in English. Convolutional Neural Networks seems to be the best performing model here and we will try to implement that extending it to Hindi-English mix text if possible.

CHAPTER 2

IMPLEMENTATION AND SYSTEM ARCHITECTURE

2.1 SYSTEM ARCHITECTURE

In this section we go through some of the basic architecture of Deep Learning Models and a few of those related closely to the task.

2.1.1 Deep Neural Networks

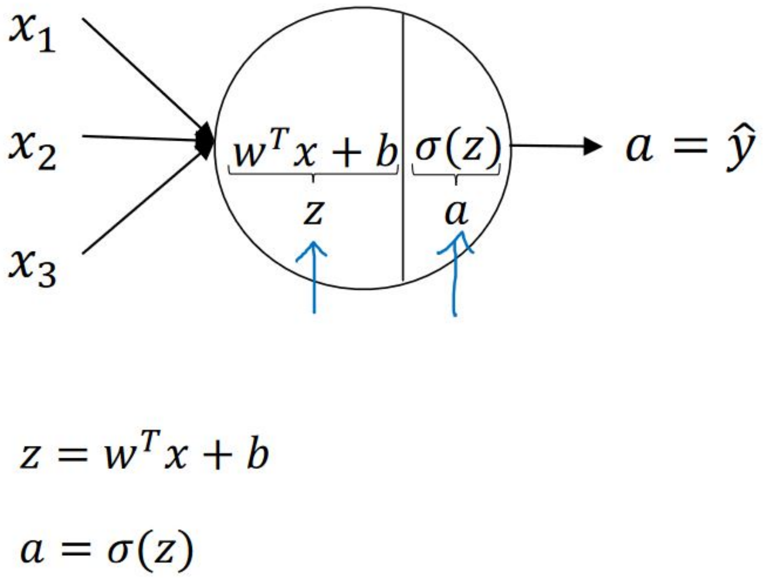
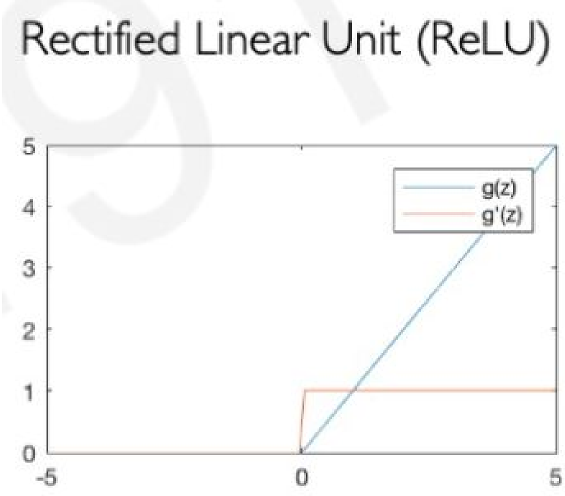


Figure 2.1: Neuron representation.

xi denote input features. *w* denotes weight corresponding to the features and *b* is bias vector to shift preference of features. *a* is Activation Function. Our Model uses ReLU (2.2) activation function in the perceptron. Standard NN is made using a



$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

Figure 2.2: ReLU.

hidden layer consisting of inter-related input features and nodes in hidden layer i.e. combination of multiple Neurons linked in a complete graph fashion. Deep Neural Network is extension of Standard NN (shown in 2.3) that contains multiple hidden layers. Summation of difference of output, *y*, from the actual output is called Loss and function that calculates this is using the input features, weights and activation function is called Loss Function. While training a Deep Neural Model multiple iterations are carried out to get the values of *w* so that Loss function is minimal. Based on this very fundamental yet complex ideology we proceed using different techniques to identify best fit model for our task.

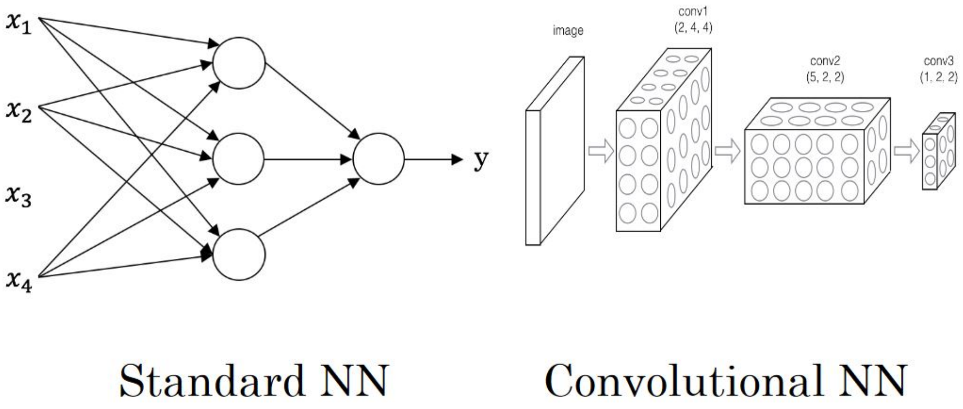


Figure 2.3: Neural Network representation.

2.1.2 Natural Language Processing

Natural language preparing (NLP) is a subject of software engineering—explicitly, a part of man-made consciousness (AI)— concerning the capacity of PCs to get text and verbally expressed words in the very way that people can. Computational semantics—rule-based human language demonstrating—is joined with factual, AI, and profound learning models in NLP. These advancements, when utilized together, permit PCs to deal with human language as text or discourse information and 'comprehend' its full significance, including the speaker or author's purpose and notion. A few NLP exercises assist the machine with getting what it's retaining by separating human content and discourse input in manners that the PC can comprehend. The following are some of these responsibilities:

- Speech recognition
- Part of speech tagging
- Sentiment analysis

Sentiment Analysis is closely related to our task that makes understanding and usage of NLP in the project crucial.

2.1.3 Bag of Words Model

The BoW model, is a strategy for separating text credits for demonstrating, for example, ML methods. The technique is clear and versatile, and it very well might be utilized to extricate data from records in an assortment of ways. A Bag of Words is a book portrayal that depicts the presence of words shows up in an archive. It consists of 2 things:

1. Vocabulary.
2. Occurrence of words.

This strategy discards all the information regarding document's structure of word order, and that's the reason why it is called "bag" of words. The model essentially thinks often about whether perceived terms show up in the report, not where they show up.

2.2 METHODOLOGY

2.2.1 PreProcessing Dataset

Thousands of tweets are used for dataset purpose using Twitter API. We used xxatp to de-identify person occurrences (e.g. @someone), xxurl to de-identify url occurrences, xxrtm to de-identify source of modified retweets, and xxrtu to de-identify source of unmodified retweets. In a word, we eliminated the frequent irractional characters (for example - <unk>,<br/>, etc) and kept the repeated chars (e.g. poooor). To replace Hexa-decimal ESC sequences by the char they represent, we utilized HTML unescape. PreProcessing was done thoroughly, which come about in the removal of punctuation and URLs, as well as the replacement of user names and emoticons [5].

2.2.2 Word Embedding

Embedding-models utilize the distributional trademark that a word is characterized by the nearby words it has to evaluate semantic similarities between words. The models quantify semantic characteristics of words by relating nearby words near to one another in an Euclidean-space. The models can efficiently build a efficient and quality word-embedding from the company of terms in a large corpus if the corpus is large enough. Every word of the lexicon is transformed into a vector of real numbers via word embedding. Continuous BoW and Skip Gr@m (Fig. 2.4) are two common feed-forward neural network-based word embedding models presented by Mikolov et al [6]. In embedded models, a fixed-size sliding-window advances along with the corpus text. Let the word in the middle of the sliding window be the active word W(i), and the words in neighbours contained in the sliding-window be context-words c for a given location of the sliding window. The continuous BOW model estimates the active word W(i) by

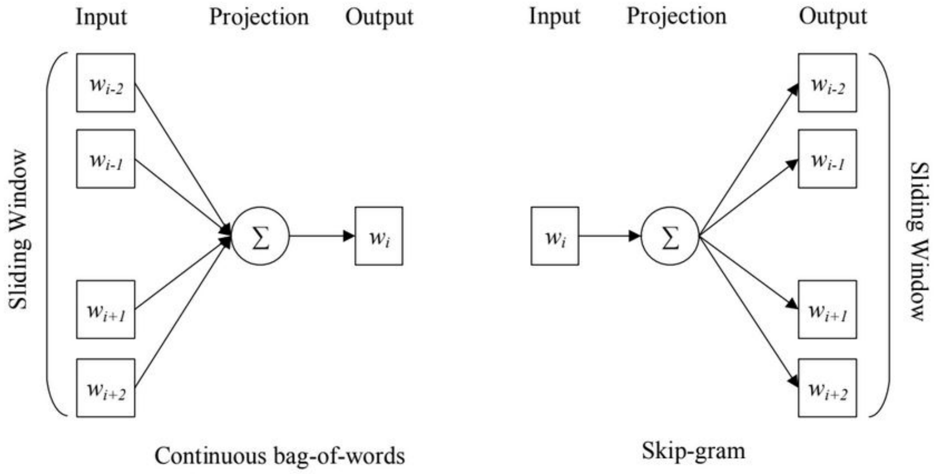


Figure 2.4: Sliding Window and skip-gram.

the neighbouring c(context-words), that is P(W*i*|c). Whereas, the skip gram method utilizes the active word W(i) to estimate the neighbouring words c, that is, P(c|W*i*). Suppose there is a segment of sentence A B C D E. In ongoing Bag Of Words (BoW), the neighbouring words A,B,D,E is utilized to estimate the center word c, in contrast, in skip gram, current word C is utilized to estimate the neighbouring words A,B,D,E. Model training's goal is to identify a word-embedding which optimizes P(c|Wi) or P(Wi|c) throughout a collection. In every iteration of model-training, every word is (1) gets nearer to the neighbouring words (2) gets away from the word that are not in the context.

CHAPTER 3

RESULTS AND DISCUSSION

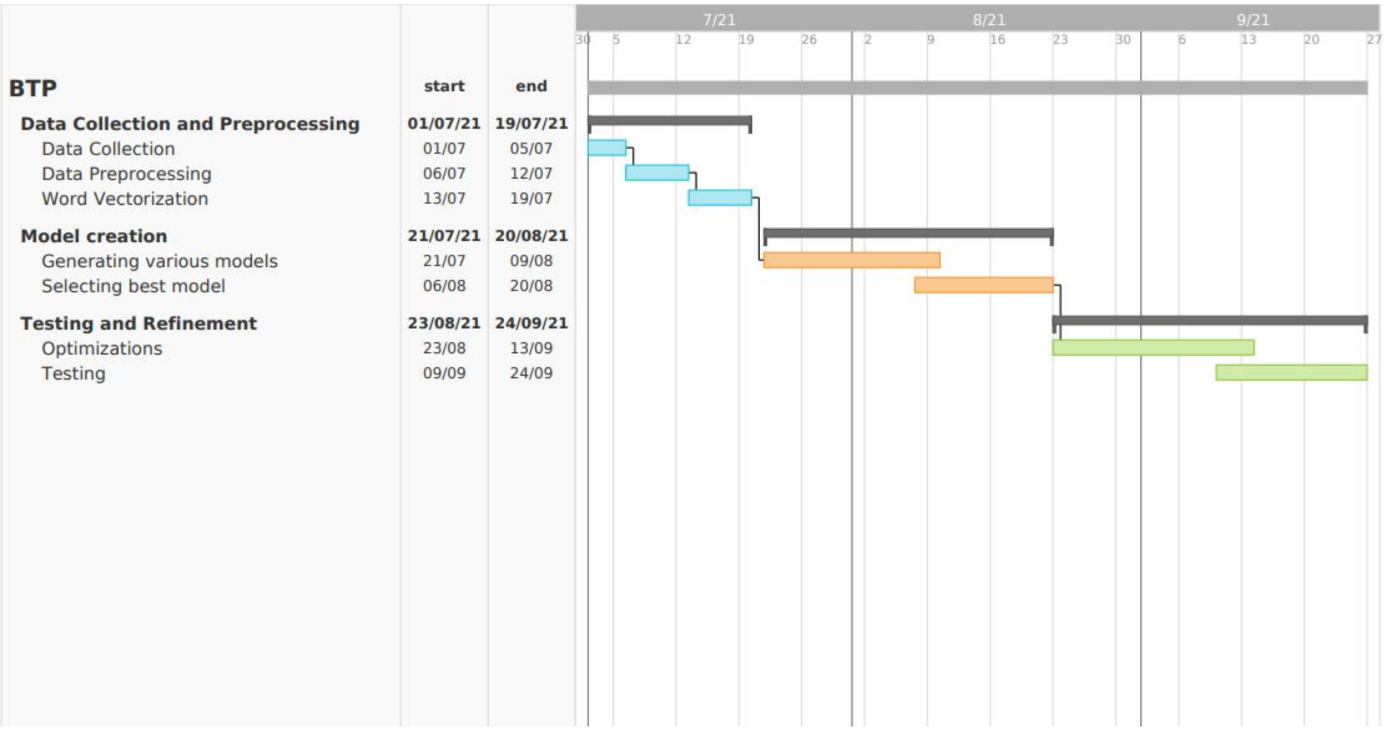
3.1 PROGRESS MADE SO FAR

We have formulated different models for hate speech detection step by step. As mentioned above this is a binary classification problem Logistic regression is used initially but it doesn't perform well. Then techniques of Natural Language Processing are adopted and we get better results. As the task can be closely related to sentiment analysis of text, similar architecture and methodology are deployed. Dataset are tweets collected using Twitter API that are processed as described in previous sections. These annotated dataset give input feature and desired output distinctively. As of now Bag-of-Words Model is implemented and Naive Bayes Model is trained on training set.

3.2 FURTHER TASKS

Plan is to transition this model towards Convolutional Neural Networks (CNN) Model, that seems to be a better choice with higher accuracy based on the research done previously [7] (mentioned in Literature Survey). Further we aim to include Hindi-English text and extend the task to detecting hate speech in Hinglish or Hindi text.

3.3 Gantt Diagram



BTP	start	end
Data Collection and Preprocessing	**01/07/21**	**19/07/21**
Data Collection	01/07	05/07
Data Preprocessing	06/07	12/07
Word Vectorization	13/07	19/07
Model creation	**21/07/21**	**20/08/21**
Generating various models	21/07	09/08
Selecting best model	06/08	20/08
Testing and Refinement	**23/08/21**	**24/09/21**
Optimizations	23/08	13/09
Testing	09/09	24/09

Figure 3.1: Gantt Diagram.

REFERENCES

[1] S. Malmasi and M. Zampieri, “Detecting hate speech in social media,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, (Varna, Bulgaria), pp. 467–472, INCOMA Ltd., Sept. 2017.

[2] Y. Kim, “Convolutional neural networks for sentence classification,” 2014.

[3] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.

[5] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, “A dataset of Hindi-English code-mixed social media text for hate speech detection,” in *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, (New Orleans, Louisiana, USA), pp. 36–41, Association for Computational Linguistics, June 2018.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” 2013.

[7] M. A. Bashar and R. Nayak, “Qutnocturnal@hasoc’19: Cnn for hate speech and offensive content identification in hindi language,” 2020.